Quiz 1b

1. (5 points) In the spirit of new-if, we decide to write new-and.

(define (new-and x y z)

(and x y z))

1. (0.5 points each) How many times is + invoked in the following code:

(new-and (+ 1 1) #f (+ 2 2))

* In applicative order? \_\_\_\_\_\_\_\_\_
* In normal order? \_\_\_\_\_\_\_\_\_
* In Scheme? \_\_\_\_\_\_\_\_\_

1. (0.5 points each) What will Scheme output? If it is an error, write “Error”.

* (new-and 2 'foo (/ 1 0))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* (and 2 'foo (/ 1 0))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* (new-and 'neil 'patrick 'harris)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (2 points) Are there any expressions where replacing and with new-and will change what happens? If so, give an example and explain why there is a difference.
2. (5 points)
   1. (2 points) Write a procedure nth-letter that takes a word and a number n and returns the nth letter in that word. You can assume that n will be between 1 and the length of the word, inclusive.

>(nth-letter 'hello 1)

h

>(nth-letter 'hello 5)

o

* 1. (3 points) Write a function nth-every using nth-letter that takes a sentence of words and sentence of numbers. nth-every will perform nth-letter on every member of the sentence of words using the corresponding number from the sentence of numbers and put them in a word. Assume both sentences will be the same length.

>(nth-every ‘(hi hello hola) ‘(1 1 1))

hhh

>(nth-every ‘(hi hello hola) ‘(1 2 4))

hea